別紙 1

論文の内容の要旨

論文題目	深さ優先ヒューリスティック探索による ソフトウェアモデル検査効率化
学位 申請者	前岡　淳

第1章においては論文の概要および構成が示されている．ソフトウェアモデル検査は，プログラムにおいてテストでは検出が難しい不具合を検出することができる重要な技術である．一方，複雑で膨大となる並行プログラムの状態遷移を探索するため，性能向上が大きな課題である．この課題に対し本研究では，従来とは異なる探索手法を提案し，代表的なソフトウェアモデル検査ツールJava PathFinder（JPF）に組み入れた．実験により，多くの場合において従来手法よりも早く不具合を検出できることが示された．加えて同ツールにおいて，より様々な性質を検証するため，線形時相論理（LTL： Linear Temporal Logic）の検証やその際に公平性条件を扱う拡張も行い，その際にも提案手法を活用した効率化を行った．以上のように本研究では，重要な課題に対して，従来とは異なるアイディアを提案，展開することにより，有効な技術を構築し評価している．

本論文の第2章においては，背景となるソフトウェアモデル検査に関する論文の概要および構成が示されている．モデル検査は，対象とするモデルが取りうる状態を探索し，モデルが満たすべき性質を満たすかどうかを調べることで不具合を検出する技術である．そのために，並行プログラムにおける複数スレッドの実行順序など，非決定的要因についての様々な選択肢を効率よく探索することが重要である．近年では，実装コードそのものを扱うソフトウェアモデル検査が注目されている．本論文において扱うJPFは，Javaプログラムを対象としたソフトウェアモデル検査器である．

第3章においてはソフトウェアモデル検査における既存の探索アルゴリズムとその課題を論じている．モデル検査においては，モデルの規模に応じて探索すべき状態数が指数的に増加し，現実的な時間で検証が終了しないという状態爆発の問題がある．特にソフトウェアモデル検査では，プログラムという抽象度の低いモデルを対象とするため，この状態爆発の問題が顕著である．そこで，この課題を解決する手法として最良優先探索が提案され，有効性の高いものとしてよく用いられている．最良優先探索においては，探索の候補となる状態の先に

不具合の含まれる見込みが高いと思われる状態から優先的に探索を進めることにより，不具合を早く発見することを目指す．優先度のつけ方に関しては，例えば「スレッドの切り替えを多く起こすようにする」といった，いくつかの代表的なポリシーが提案され，活用されている．

しかし，最良優先探索においては第一に，現在の探索候補状態を保持し検討するためのメモリ使用量が多く，あふれた候補を捨てることも発生しうる．第二に，深い状態，すなわち実行が進んだ状態に到達するのに時間がかかることも多く，その結果不具合発見が遅くなる場合がある．最後に，より様々な性質の検証を行うためのLTL検証には適用できない．これらの点はいずれも，最良優先探索が幅優先探索に近い挙動を持つことに起因する．

第4章においては提案手法である深さ優先ヒューリスティック探索（DFHS: Depth First Heuristic Search）について説明している．従来手法である最良優先探索は各状態に対して優先順序づけを行うのに対して，提案するDFHSは，不具合に至る見込みが低いと判断した枝の探索を打ち切る枝刈りを行う．このため，探索候補状態を大量に保持する必要がなく，またより早く深い状態に到達する可能性がある．この点はあるポリシーを実現する際の実装の違いであり，最良優先探索のときと同等のポリシーに基づく探索をDFHS手法においても実現することができる．さらに，枝刈りに加えて，枝の選択順序を制御することで，最良優先探索同様の考え方を部分的に活用することもできるようになっている．

提案するDFHSは深さ優先探索を基本としているため，LTL検証の実装に組み合わせることも可能である．望ましくない状態に到達しないという性質（安全性）の検証においては，個々の状態に対し特定の性質を満たすかを判定する．一方，望ましい状態に到達するという性質（活性）の検証においては，実行パスに対し特定の性質を満たすかを判定することになる．LTLは活性を含む様々な性質を表現可能な論理系である．LTL検証を扱う場合，深さ優先探索に基づき安全性検証よりもさらに大きな状態空間の探索を行うことになる．ここで提案するDFHSを用いることで効率化の恩恵を得ることができる．提案手法においては，LTL検証の際によく用いられる公平性条件（特定のスレッドが無限に選ばれる状況を無視して検証を行う）を効率的に扱うための最適化も組み込んでいる．

第5章においては提案したDFHS手法のJPFにおける実装について説明している．既存のJPF実装における探索制御部を拡張することにより，上記の機構を組み込むことができている．

第6章においては提案に関する評価実験について述べている．実験においては主に，従来の最良優先探索と提案するDFHSに関し，同等のポリシー（単一または複数）を用いた際の不具合発見時間や状態数などを評価した．ここで複数のポリシーを用いる場合は，クラウドを用いて複数のポリシーを用いた検査を同時に走らせる場合など実用シナリオを想定している．これは，対象とするプログラムやそれに含まれる未知の不具合に対して，どのポリシーが有効かを事前に知ることは現実的にできないためである．実験の結果，多くの場合において提案したDFHSが早く不具合を発見することができることが示された．そのほか，DFHS上でのさらなる工夫である枝の選択順序制御や，LTL検証および公平性条件の扱いの効率化についても評価を行い，提案の有効性が示された．

第7章においてはまとめと将来課題について述べている．以上の研究成果についてまとめ，今後のさらなる活用に向けたアイディアを示している．

別紙２

論文審査の結果の要旨

学位申請者氏名　前岡　淳

審査委員主査　大須賀　昭彦

委員　石川　冬樹

委員　田中　健次

委員　多田　好克

委員　田原　康之

以下においてはまず，各章ごとの内容と審査結果について述べる．

第1章においては，本論文の内容について概観している．この章においては，各章の内容が端的に，適切にまとめられている．

第2章においては，本論文の背景および前提知識となるソフトウェアモデル検査に関して記している．モデル検査の特徴および利点と課題，前提知識となるLTL（Linear Temporal Logic）やJPF（Java PathFinder）ツール，および関連研究について適切に記されている．

第3章においては，従来の探索アルゴリズムについて論じている．実験においても比較対象とされている深さ優先探索，最良優先探索およびLTL検証アルゴリズムのそれぞれについて，その仕組みが図および擬似コードを交えて明確に示されている．特に比較対象となる最良優先探索については，その課題が論じられ明確に示されている．

第4章においては，提案手法である深さ優先ヒューリスティック探索（DFHS: Depth First Heuristic Search）について説明している．基本となる枝刈り機能について擬似コードを交えて定義し，その上で実現される各ポリシーについて図を用いてわかりやすく説明している．続いて，最良優先探索同様の考え方を部分的に活用する枝の選択順序最適化，およびLTL検証における公平性条件の扱いの効率化についても，擬似コードおよび図を用いて明確に説明している．

第5章においては，提案手法のJPFツールへの組み込みについて述べている．第4章で示した機能のそれぞれについて，JPFの探索制御部への組み込み方法を実際のJavaコードも用いて明確に示している．

第6章においては，提案手法の評価実験について述べている．実験においては主に，従来の最良優先探索と提案するDFHSに関し，不具合発見時間や状態数などを評価している．これらの探索の有効性は，対象とするプログラムとその不具合，およびそれに対して用いるポリシーに依存する．このため25個のベンチマークプログラムを用い，従来手法と提案手法で同一のポリシー単数を用いる場合，同一のポリシー複数を用いる場合（実用シナリオを想定）など，多くの状況を想定し注意深く実験を行っている．実験の結果，多くの場合において提案したDFHSが早く不具合を発見することができることが示された．そのほか，DFHS上でのさらなる工夫である枝の選択順序制御や，LTL検証および公平性条件の扱いの効率化についても評価を行い，提案の有効性が示された．以上のように，多くの実験を通して注意深く提案手法の有効性および有用性を論じている．

第7章は本論文のまとめであり，本論文で示された技術の有効性や有用性，および今後の活用に向けて考えられるアイディアについて簡潔に記されている．

以上のように，本論文は，ソフトウェアモデル検査の効率化という重要な課題に取り組んでいる．これに対して，従来とは異なるアイディアに基づき提案手法DFHSを適切に定めた．また枝の選択順序制御や，LTL検証および公平性条件の扱いの効率化とそのアイディアを展開することにより，有効な一連の技術を構成した．またその有効性および有用性について，多くの実験を通して評価した．これらの点について，論文および発表にしっかりとまとめられており，質疑においても的確な応答がなされた．

よって本論文は博士（工学）の学位論文として十分な価値を有するものと認める．